

INFORMATION PROCESSOR, METHOD FOR
PROCESSING INFORMATION AND MEMORY MEDIUM
FOR STORING PROGRAM READABLE BY COMPUTER

5 BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to an information processor for generating printing data to be transmitted to a printing device, a method for processing information and a memory medium, and more particularly to a technique for composing one print job by a plurality of print requests in a host computer.

Related Background Art

Conventionally, in the case where one document (document data) generated in an application is printed in the host computer, a set of print setting attributes composed of a plurality of print setting items previously set by a printer driver has been ordinarily defined for one print job generated from the document.

There has been known software having a "collective printing" function by which the document data instructed to be printed by the application is not outputted to a printer and held in a client, that is to say, a host computer and a plurality of document data is simultaneously transmitted to the printer as one job.

In the case where a user has hitherto wished to

00699389 103100

simultaneously print the print jobs generated by, for instance, an application A suitable for preparing documents, an application B suitable for preparing charts or tables, an application C suitable for preparing drawings, etc., however, the "collective printing" function has been disadvantageously inferior in its maneuverability, because the "collective printing" function only serves to collect a plurality of print jobs together to one print job, and, because upon printing data, the data is outputted in regular order according to which the data is spooled for the collective printing in response to the instruction of collective printing, so that the user must spool the data on outputted pages. For instance, in the case where the user wishes to print data composed of a document, a drawing, a document, a table and a document each of which is written on one page, the user has inconveniently needed to prepare a first page by the application A and spool it, prepare a second page by the application B and spool it, prepare a third page again by the application A and spool it, prepare a fourth page by the application C and spool it and prepare a fifth page by the application A and spool it.

Further, in the case where the user wishes to see the preview of data collected together as one data to be printed, the user can see the preview provided by each application, however, it has been impossible for

001E01" 68E66960

the user to see the preview based on a plurality of print settings set by the printer driver.

Still further, in the case where the user wishes to put a plurality of files together and print them as one print job, when the user tries to carry out a designation of finishing relative to the print job thus obtained, there is no means for providing the designation of finishing relative thereto. There is undesirably provided no means having a preview function either. Therefore, unless the user prints the data, the user cannot disadvantageously understand how the data is actually printed and outputted by the printer.

Still further, in the case where the "collective printing" is employed to use a page layout function (N pages/sheet) in a device side, the same page layout processing can be carried out relative to a plurality of jobs, however, the print jobs cannot be combined together while a different layout is applied to each of the original print jobs.

As described above, the conventional "collective printing" only functions as a continuous printing, and therefore, one print job cannot be prepared by using a different layout for each page.

Further, as described above, according to the prior art, only one page layout can be provided for one job. Therefore, in the case where one job is formed for the collective printing, the user cannot designate

the page layout when the print jobs are previously
outputted from the applications for the collective
printing, that is to say, printing the print jobs in
one lot, so that the user has needed to select only one
5 page layout upon collective printing.

SUMMARY OF THE INVENTION

Thus, it is an object of the present invention to
provide an arrangement for combining together print
10 jobs respectively having print setting information.

Further, when one composed job is formed by
combining together a plurality of print jobs
respectively having the print setting information, if
the print jobs are respectively separately set for
15 printing operation, any mechanism will be needed for
unifying a plurality of settings. Accordingly, it is
another object of the present invention to provide an
arrangement by which a user can understand what is
changed to unify the settings upon automatic
20 unification.

Still further, it is a still another object of the
present invention to provide a function for managing
and editing functions for each job after a plurality of
print jobs are put together, for instance, a function
25 for rearranging a print order for each job irrespective
of an order in which the print jobs are printed and a
function for deleting the print jobs from a composed

09699389 103100

job for each job or returning the composed job to a plurality of original single jobs.

5 Still further, it is a still another object of the present invention to provide an arrangement by which the print jobs can be edited on each logical page, for instance, unnecessary pages can be deleted.

10 Still further, it is a still another object of the present invention to provide an arrangement by which the composed job is returned to original single jobs after the composed job is edited on each page.

15 Furthermore, by taking the above described problems into consideration, it is a further object of the present invention to combine together print jobs to which page layout designations are respectively applied, while the page layout designations are maintained.

20 In addition, it is a further object of the present invention to unify layouts relative to a composed job having a plurality of page layouts for each physical page.

25 For attaining the above described objects, an information processor according to the present invention has such a configuration as described below. Specifically, an information processor for generating printing data to be transmitted to a printing device comprises a spooler for converting data to be printed which is generated by an application into print jobs

09699389 103100

and temporarily storing the print jobs; a composition
instructing unit for instructing a plurality of print
jobs corresponding to the different data to be printed
to be combined together so as to generate one composed
5 job; and a setting unifier for analyzing the print
setting information of a plurality of print jobs when
the composition instructing unit instructs the
plurality of print jobs to be combined together so as
to obtain one composed job, and generating print
10 setting information for the composed job by unifying
information which can be respectively merely set to one
print job.

Further, the setting unifier further includes a
recognizing unit for analyzing the print setting
15 information of a plurality of print jobs to be combined
together and recognizing to select whether the settings
are unified or the print jobs are not combined together
when information which can be set only to one print job
is mutually different.

20 Still further, the spooler converts the data to be
printed into the print jobs of intermediate code format
and temporarily stores the print jobs as page
description files for each page.

Still further, information for designating the
25 page description files laid out on a physical page is
added to the print setting information of the composed
job.

00609389 103100

Still further, the above described print setting information is temporarily stored as a print setting file of each print job.

5 Still further, the information processor further comprises a preview display controller for controlling a preview based on the print setting information of the print jobs or the composed job to be displayed.

10 Still further, the information processor further comprises an order controller for operating a plurality of print jobs in the composed job to reshuffle the order of the print jobs.

15 Still further, the information processor further comprises a job cancelling unit for operating a plurality of print jobs in the composed job to cancel a specific print job.

Still further, the information processor further comprises a job divider for dividing the composed job into a plurality of print jobs before they are joined together.

20 Still further, the information processor further comprises a job duplicating unit for designating the print job or the composed job to prepare the duplication of the designated print job.

25 Still further, the print job or the composed job further includes a setting initializing unit for returning the intermediate code format as the base of the job to an initial state upon preparation of the

09699389-103100

Still further, the information processor further comprises a page editing unit for cancelling a page designated relative to a logical page in the print job or the composed job.

Still further, the information processor further comprises a description instruction generator for converting the data of the intermediate code format temporarily stored by the intermediate code converter into a description instruction which can be interpreted by the description unit of an OS for outputting; and a print instruction allocator for sending the print instruction received through the description unit of the OS from the application to the intermediate data converter and sending the print instruction received through the description unit of the OS from the description instruction generator to the printing data generator.

In addition, the description instruction is a GDI function, the print instruction is a DDI function and the printing data is a printer language.

Furthermore, the information processor further comprises a layout unification instructing unit for instructing the layout information of the composed job to be unified and the composed job information generator unifies the layout information of the composed job by all physical pages when the layout unification instructing unit instructs the layout information to be unified.

Besides, the present invention can be realized by a method for controlling the image processor, a memory medium in which a program is stored or a computer program.

Other features and advantages of the present invention will be apparent from the following description taken in conjunction with the accompanying drawings, in which like reference characters designate the same or similar parts throughout the figures thereof.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram for explaining the configuration of a print controller showing an embodiment of the present invention;

Fig. 2 is a block diagram showing a representative print system of a hostcomputer to which a printer is connected;

Fig. 3 is a block diagram showing the

09699389 103100

configuration of the print system for temporarily
spooling an intermediate code before a print
instruction from an application is converted into a
printer control command;

5 Fig. 4 is an explanatory view of the printer in
the present invention;

Fig. 5 is a flowchart showing processings in a
spooler 302;

10 Fig. 6 is a flowchart showing a print control in a
spool file manager 304;

Fig. 7 is a flowchart showing processings in a
despooler 305;

Fig. 8 shows one example of a print setting
screen;

15 Fig. 9 shows one example of a print spool setting
screen;

20 Fig. 10 is a view showing one example of a data
format delivered when the spool file manager 304
supplies a print request for a physical page to the
despooler 305;

Fig. 11 is a view showing one example of a data
format delivered when the spool file manager 304
supplies a print request for a physical page to the
despooler 305;

25 Fig. 12 is a view showing one example of a data
format delivered when the spool file manager 304
supplies a print request for a physical page to the

00699389-103100
DOTED-6866960

despooler 305;

Fig. 13 is a view showing one example of a data
format delivered when the spool file manager 304
supplies a print request for a physical page to the
despooler 305;

Fig. 14 is a view showing one example of a data
format delivered when the spool file manager 304
supplies a print request for a physical page to the
despooler 305;

Fig. 15 is a flowchart showing setting change
processes in a setting change editor 307;

Fig. 16 shows one example of a screen displaying a
print job list spooled in the spool file manager 304;

Fig. 17 shows one example of a screen of a
previewer 306;

Fig. 18 shows one example of a screen of the
setting change editor 307;

Fig. 19 is a flowchart showing the operation of a
user when the print jobs are spooled in the spool file
manager 304 as shown in Fig. 16;

Fig. 20 is a flowchart showing the operation of
the user upon composing the jobs together;

Fig. 21 is a conceptual view for explaining the
unification and rounding off of settings when a
plurality of documents are combined together;

Fig. 22 shows one example of a user interface for
selecting a plurality of documents to be combined

09699389 103100

Figs. 31A and 31B show one example of data processings such as reshuffling the order of the

Fig. 32 is a flowchart showing one example of job editing processes in the spool file manager 304;

Fig. 34 shows a flowchart showing one example of processes in the spool file manager 304 upon cancelling the print jobs for each page by the setting change editor 307;

Fig. 36 shows one example of a processing flow for changing a layout in the setting change editor 307;

Fig. 38 shows one example of a processing flow for calculating physical page information;

Fig. 40 shows one example of a screen of the setting change editor 307; and

Fig. 41 shows one example of a screen of the setting change editor 307.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Now, preferred embodiments to which the present invention is applicable will be described below.

Fig. 1 is a block diagram for explaining the configuration of a printer control system showing an embodiment of the present invention. The present invention is applicable to any printer control system by which the functions of the present invention can be executed, such as a system composed of a single device or a plurality of devices, a system which is connected through a network such as LAN, WAN, etc. to perform various processings.

In Fig. 1, a host computer 3000 is provided with a CPU 101 for executing a document processing in which graphic forms, images, characters, tables (including table calculations), etc. are mixed on the basis of a document processing program or the like stored in the ROM for program of a ROM 103 or an external memory 111. The CPU 101 generally controls respective devices connected to a system bus 104. In the ROM for program of the ROM 103 or the external memory 111, an operating system program (called an OS, hereinafter) as a control program of the CPU 101 is stored. In a ROM for font of the ROM 103 or the external memory 111, font data employed for a document processing is stored. In a ROM for data of the ROM 103 or the external memory 111, various kinds of data used for carrying out the

A keyboard controller (KBC) 105 controls a key input from a keyboard 109 or a pointing device not shown. A CRT controller (CRTC) 106 controls the display of a CRT display (CRT) 110. Reference numeral 107 designates a disk controller (DKC) and controls an access to the external memory 111 such as a hard disk (HD), a floppy disk (FD), etc. for storing a boot program, various kinds of applications, font data, a user file, an edit file, a printer control command generating program (called a printer driver, hereinafter), etc. A printer controller (PRTC) 108 is connected to a printer 1500 through a two-way interface 121 to control a communication with the printer 1500.

The printer 1500 is controlled by a CPU 112. The

printer CPU 112 outputs an image signal as output information to a printer engine 117 connected to a system bus 115 on the basis of a control program stored in the ROM for program of a ROM 113 or a control program stored in an external memory 114 or the like. Further, in the program ROM of the ROM 113, the control program of the CPU 112 is stored. In the ROM for font of the ROM 113, font data or the like for generating the output information is stored. In the ROM for data of the ROM 113 is stored information utilized on the host computer, in the case of a printer having no external memory 114 such as a hard disk.

The CPU 112 can communicate with the host computer through an input unit 118 so as to inform the host computer 3000 of the information or the like in the printer. A RAM 119 is a RAM which functions as the main memory or the work area of the CPU 112 and is designed to expand a memory capacity by an option RAM connected to an extended port not shown. The RAM 119 is employed for an output information expanded area, an environmental data stored area, an NVRAM, etc. The access of the above described external memory 114 such as the hard disk (HD), an IC card, etc. is controlled by a memory controller (MC) 120. The external memory 114 is connected as an option to store font data, an emulation program, form data, etc. Reference numeral 1501 denotes an operation panel on which switches for

00699389 103100

The number of the above described external memory 114 is not limited to one and a plurality of external memories may be provided. A plurality of external memories, in which option cards as well as an integrated font and programs which can interpret printer control languages different in their language system are stored, may be connected together. Further, an NVRAM not shown may be provided so as to store printer mode setting information from the operation panel 1501.

Fig. 2 is a block diagram showing a representative printing process in the host computer to which a printing device such as the printer is directly connected or via a network. An application 201, a graphic engine 202, a printer driver 203 and a system spooler 204 are program modules with which the RAM 102 is loaded and which are executed by the OS or modules using its module, when they exist as files stored in the external memory 111 and are executed. Further, the application 201 and the printer driver 203 can be added to the HD of the external memory 111 via the FD of the external memory 111, a CD-ROM not shown or a network not shown. The application 201 stored in the external memory 111 with which the RAM 102 is loaded is executed by the RAM 102. When the printing operation is carried

out on the printer 1500 from the application 201, an output (image description) is performed by using the graphic engine 202 with which the RAM 102 is similarly loaded to be executable.

5 The graphic engine 202 similarly loads the RAM 102 with the printer driver 203 prepared for each printing device from the external memory 111 and sets the output of the application 201 to the printer driver 203.

10 Then, the graphic engine 202 converts a GDI (Graphic Device Interface) function received from the application 201 into a DDI (Device Driver Interface) function and outputs the DDI function to the printer driver 203. The printer driver 203 converts the DDI function received from the graphic engine 202 into a control command, for instance, a PDL (Page Description Language) which can be recognized by the printer. The printer control command thus obtained is adapted to be outputted to the printer 1500 as printing data through the interface 121 via the system spooler 204 with which
15 the RAM 102 is loaded by the OS.
20

 The print system according to the present embodiment comprises a configuration in which the printing data from the application is temporarily spooled by intermediate code data as shown in Fig. 3 in
25 addition to the print system having the host computer and the printer as shown in Fig. 2.

 A system shown in Fig. 3 is formed by extending

09699339 "103100

the system shown in Fig. 2, the application 201 is released from the printing process when the printer driver 203 completely converts all the print instructions from the graphic engine 202 into the control commands of the printer. On the other hand, in the system shown in Fig. 3, the application is released from the printing process when a spooler 302 converts all the print instructions into the intermediate code data and outputs the intermediate code data to the spool file 303. The latter ordinarily takes a shorter time than the former. Further, in the system shown in Fig. 3, the contents of the spool file 303 can be processed. Thus, can be realized functions which are not provided in the application, such as enlargement/reduction, reduction and print of a plurality of pages to one page, etc. relative to the printing data from the application.

For achieving these objects, the system has been adapted to be extended in such a manner that the printing data is spooled by the intermediate code data as shown in Fig. 3 as compared with the system shown in Fig. 2. In order to process the printing data, a setting is ordinarily carried out from a window

provided by the printer driver 203. The printer driver 203 stores the contents of the setting on the RAM 102 or in the external memory 111.

Now, the detail of the print system shown in Fig. 3 will be described below. As shown in Fig. 3, according to the extended processing system, a dispatcher 301 receives the DDI function as the print instruction from the graphic engine 202. In the case where the print instruction (DDI function) that the dispatcher 301 receives from the graphic engine 202 is based on the print instruction (GDI function) sent to the graphic engine 202 from the application 201, the dispatcher 301 loads the RAM 102 with the spooler 302 stored in the external memory 111 and transmits the instruction (DDI function) not to the printer driver 203 but to the spooler 302.

The spooler 302 analyzes the received print instruction, converts it into the intermediate code for each page and outputs the intermediate code to the spool file 303. The spool file of the intermediate code stored for each page is called a page description file (PDF). The spooler 302 obtains from the printer driver 203 the processed settings of the printing data set to the printer driver 203 (Nup, duplex printing, stapling, color/monochrome designations, etc.) and stores them in the spool file 303 as files for each job. The setting file stored for each job is called a

00699389-103100

job setting file (also called simply a SDF: Spool Description File). The job setting file will be described later. The spool file 303 is generated on the external memory 111, however, it may be generated on the RAM 102. Further, the spooler 302 loads the RAM 102 with a spool file manager 304 stored in the external memory 111 and informs the spool file manager 304 of the status of generation of the spool file 303. Then, the spool file manager 304 decides whether or not a printing operation can be performed in accordance with the contents of the processed setting of the printing data stored in the spool file 303.

When the spool file manager 304 decides that the printing operation can be performed by employing the graphic engine 202, the spool file manager 304 loads the RAM 102 with a despooler 305 stored in the external memory 111 and instructs the despooler 305 to print the page description file of the intermediate code described in the spool file 303.

The despooler 305 processes the page description file of the intermediate code included in the spool file 303 in accordance with the job setting file having the process setting information included in the spool file 303 to regenerate the GDI function, and outputs again the GDI function via the graphic engine 202.

In the case where the print instruction (DDI function) which the dispatcher 301 receives from the

00699389 103100
DDI 68266960

graphic engine 202 corresponds to a print instruction based on a print instruction (GDI function) supplied to the graphic engine 202 from the despooler 305, the dispatcher 301 sends the print instruction not to the spooler 302 but to the printer driver 203.

The printer driver 203 generates a printer control command composed of a page description language or the like on the basis of the DDI function got from the graphic engine 202, and outputs the command to the printer 1500 via the system spooler 204.

Further, referring to Fig. 3, there is shown an example in which a previewer 306 and a setting change editor 307 are added to the extending system explained so far so that a print can be previewed, a print setting can be changed, and a plurality of jobs can be composed together.

In order to preview the print, change the print setting and compose a plurality of jobs together, the user needs to first designate "store" in a pull-down menu as means for "designating an output destination" in the property of the printer driver shown in Fig. 9. When the user wishes to see only the preview, the user can select a "preview" as the designation of the output destination.

Contents set in the property of the printer driver as described above are stored as setting files in a structure body (called DEVMODE, in Windows OS) provided

00509389 103100
007E07 6866960

5 The spool file manager 304 reads out the process
setting through the printer driver, and when the store
is designated, the page description file and the job
setting file are generated and stored in the spool file
303 as described above. Then, the window screen of the
10 spool file manager is popped up as illustrated in Fig.
16 and the jobs spooled in the spool file 303 are
listed up and displayed. Fig. 16 shows an example in
which four jobs are spooled. Thus, the jobs can be
operated by pushing down a menu bar or a menu icon just
15 below it. The number of operations of the menu bar is
equal to that of the menu icon. There are eleven kinds
of operations, such as "print" while the job is
selected, "print conservatively" for printing by
leaving the spool file of the intermediate code as it
20 is, "preview" for seeing the output preview of the job
by considering the print setting, "delete" for deleting
the spool file of the intermediate code, "duplicate"
for generating a copy of the spool file of the
intermediate code, "compose" for composing together a
25 plurality of jobs of the spool file of the intermediate
code to obtain one job, "divide" for dividing the
composed job into a plurality of original jobs, "job

edit" for changing the print setting (layout setting, finishing setting, etc.) of a single job or the composed job, "move to the first" for moving a job to the top in print order, "move faster by one" for moving a job faster by one in print order, "move slower by one" for moving a job slower by one in print order and "move to the last" for moving a job to the last in print order.

When the preview of a single job or the composed job is designated on the window screen (see Fig. 16) of the spool file manager, the RAM 102 is loaded with the previewer 306 stored in the external memory 111 to instruct the previewer306 to execute the preview processing of the job of the intermediate code described on the spool file 303.

The previewer 306 sequentially reads the page description files (PDF) of the intermediate code contained in the spool file 303, processes them in accordance with the contents of the process setting information included in the job setting file (SDF) stored in the spool file 303 and outputs the GDI function to the graphic engine 202. Then, the graphic engine 202 outputs description data to a client area of itself, so that the data can be outputted onto the screen.

The graphic engine 202 can perform a suitable rendering depending on a designated output destination.

00699389 103100

Accordingly, there can be realized a method in which the previewer 306 processes the intermediate codes included in the spool file 303 in accordance with the contents of the process setting contained in the spool file 303 and outputs the data thus obtained by employing the graphic engine 202, like the despooler 305. As described above, the process setting set by the printer driver is stored in the spool file 303 as the job setting file and the data of the page description file is processed on the basis of the job setting file and the processed data is outputted. Thus, can be provided for the user a print preview, as to how actual description data is printed, further print previews near those outputted by the printer respectively according to the cases where Nup (N logical pages are reduced, arranged and printed on one physical page) is designated, a duplex side print is designated, a binding print is designated and a stamp is designated. A preview function of an application software for preparing a document in the prior art serves to basically describe an image on the basis of the page setting of the application, so that the print setting of a printer driver is not effected and the user cannot recognize a preview whose print is actually outputted.

As described above, the preview processing is carried out, so that the large preview of the process

09699389.1031.00

5

10

15

20

25

change editor 307 stored in the external memory 111 to instruct the setting change editor 307 to display a current or default process setting. Then, a job setting screen as illustrated in Fig. 18 is displayed.

5 The setting change editor 307 obtains the job setting file of a job to which the "job edit" is designated from the spool file 303 to change the default values of the job setting screen shown in Fig. 18 on the basis of the setting items designated in the job setting file. Referring to an example shown in Fig. 18, in the job setting file of the job to which the "job edit" is designated, the number of copies: one, a printing method: a single side printing, stapling: no, layout: one page/sheet, etc. are designated.

10

15

Also in the setting change editor 307, the page description file of the intermediate code included in the spool file 303 is processed in accordance with the contents of the process setting contained in the job setting file stored in the spool file 303 and the processed file is outputted to its client area by using the graphic engine 202, so that a small preview on the screen shown in Fig. 18 can be outputted.

20

Further, the contents of the process setting included in the job setting file stored in the spool file 303 can be changed and corrected therein. At that time, items which can be set by the printer driver 203

25

00609389-103100

may be provided in a user interface on the setting
change editor 307 or the user interface of the printer
driver 203 itself may be accessed. As shown in Fig.
18, fractions, a printing method (single side printing,
5 duplex side printing, binding printing), stapling
(saddle finisher, etc.), page layout, arrangement
order, etc. can be designated. Further, if a "detail
setting" is pushed down, most of the items which can be
designated by the printer driver can be reset. In this
10 case, the setting concerning the grade of printing or
the printing quality such as a resolution, a graphic
mode, etc. is not allowed to be changed.

As for the items changed here, the changes thereof
are verified in accordance with a verification request
15 on the setting change editor 307 to shift a control to
the spool file manager 304. For the items whose
changes are verified, the changes of the print setting
thereof are not stored in an original job setting file
but stored in a new setting file formed for outputting
20 a job which is used for editing the job or the like.
The setting file for outputting a job will be described
in more detail after Fig. 10.

Then, if the user performs a printing operation in
accordance with the contents of the changed setting
25 like the recognition by the previewer 306, the user
issues a print request on the spool file manager 304.
The print request is transmitted to the graphic engine

00699389 103100

202 and the print instruction is sent to the printer driver 203 via the dispatcher 301 to perform the printing operation.

Further, on the window screen (see Fig. 16) of the spool file manager, it is possible to designate that a plurality of jobs are composed together to print them as one print job. The above designation can be executed for the job to the output destination of which the "store" is designated in the property of the printer driver shown in Fig. 9 similarly to the preview and the setting change.

When the user combines the print jobs together, the user initially accesses the printer driver 203 from the application 201 to select the store on the user interface as shown in Fig. 9. As described above, the print jobs are stored in the spool file 303 in accordance with this selection and the window screen (see Fig. 16) of the spool file manager is popped up as illustrated in Fig. 16. The list of the spooled jobs is displayed on the window of the spool file manager. The same operation is carried out from the application 201, so that the list of a plurality of jobs is displayed on the spool file manager 304.

Here, when the plurality of jobs are selected and the "composition" is designated, the RAM 102 is loaded with the setting change editor 307 stored in the external memory 111 to instruct the setting change

09699389 "103100

editor 307 to display the process setting of a first
job or a default on the list. Then, a composition
setting screen as shown in Fig. 18 is displayed. Here,
although the setting change editor 307 is used as the
5 composition setting screen, it is to be understood that
another module may be employed.

09699369-103100
10 The setting change editor 307 processes the page
description file of the intermediate code included in
the spool file 303 on the basis of the contents of the
process setting included in the job setting information
stored in the spool file 303, and outputs the processed
file to its own client area by employing the graphic
engine 202 relative to all the jobs designated as the
composed jobs to output the processed file on the
15 screen. At that time, the small previews of all the
jobs selected in a preview area shown in Fig. 18 can be
seen. Further, when the composed job is formed,
setting files for outputting jobs in which the job
setting files of respective single jobs are extended
20 are formed. The setting file for outputting the job is
also formed when the job is edited. One setting file
is generated for one job. In the case of one composed
job, one setting file is also formed.

25 The respective jobs can be displayed on the basis
of the process setting before they are composed
together. The process settings of the jobs may be
changed and corrected to process settings for

unification so as to display the jobs as the composed job. At that time, the items which can be set by the printer driver 203 may be provided in the user interface on the setting change editor 307 or the user interface of the printer driver 203 itself may be accessed.

As for the composed job and the changed items, the changes thereof are verified in accordance with the verification requests on the setting change editor 307, as mentioned above, to shift a control to the spool file manager304. As a result of these operations, a plurality of previously selected jobs are displayed as one composed job on the window of the spool file manager.

Then, if the user performs a printing operation in accordance with the contents of the setting change like the recognition in the previewer 306, the user supplies a print request on the spool file manager 304. The print request is transmitted to the graphic engine 202 and sent to the printer driver 203 via the dispatcher 301 to execute the printing operation.

Fig. 4 is a sectional view of a color laser printer having a duplex side printing function as one example of the printer 1500.

This printer is adapted to form an electrostatic latent image by scanning a photosensitive drum 15 by a polygon mirror 31 through a laser beam modulated by

image data for each color obtained on the basis of
printing data inputted from the host computer 3000.
The electrostatic latent image is developed with toner
to obtain a visible image, which is subjected to a
multiple transfer of all colors to an intermediate
transfer body 9 to form a color visible image. Then,
the color visible image is transferred to a
transferring material 2 to fix the color visible image
thereon. An image forming part which carries out the
above described control comprises a primary charging
part including a drum unit having the photosensitive
drum 15 and a contact charging roller 17, a cleaning
part, a developing part, the intermediate transfer body
9, a sheet cassette 1, a sheet feeding part including
various kinds of rollers 3, 4, 5 and 7, a transfer part
including a transfer roller 10 and a fixing part 25.

The drum unit 13 comprises the photosensitive drum
(photosensitive body) 15 and a cleaner container 14
having a cleaning mechanism serving as the holder of
the photosensitive drum 15 which are integrally formed.
The drum unit 13 is attachably and detachably supported
by a printer main body and easily replaced by another
drum unit to meet the life of the photosensitive drum
15. The photosensitive drum 15 is configured by
applying an organic photoconductive layer to the outer
periphery of an aluminium cylinder and rotatably
supported by the cleaner container 14. The

The driving motor rotates the photosensitive drum 15 counterclockwise in accordance with an image forming operation. The surface of the photosensitive drum 15 is selectively exposed to form an electrostatic latent image. In a scanner part 30, the modulated laser beam is reflected by the polygon mirror rotating in synchronization with the horizontal synchronizing signal of an image signal by a motor 31a and the reflected beam is applied to the photosensitive drum through a lens 32 and a reflector 33.

The black developing device 21B is detachably attached to the printer main body. The color

The black developing device 21B is detachably attached to the printer main body. The color

The sleep 21BS of the black developing device 21B is arranged with a very small clearance of, for instance, about 300 μm from the photosensitive drum 15. The black developing device 21B carries toner by a feeding member incorporated in the developer and applies an electric charge to the toner by a frictional electrification so as to apply the toner to the outer periphery of the sleep 21BS rotating clockwise by the application blade 21BB. A developing bias is applied to the sleep 21BS so that a developing operation is carried out to the photosensitive drum 15 in accordance with the electrostatic latent image to form a visible image of the black toner on the photosensitive drum 15.

The three color developing devices 20Y, 20M and 20C rotate together with the rotation of the developing rotor 23 at the time of forming an image, so that the prescribed sleeps 20YS, 20MS and 20CS are opposed to the photosensitive drum 15 with a very small clearance of about 300 μ m spaced therefrom. Thus, the prescribed developing devices 20Y, 20M and 20C stop at developing positions opposed to the photosensitive drum 15 to form a visible image on the photosensitive drum 15.

At the time of forming a color image, the developing rotor 23 rotates for each rotation of the

intermediate transfer body 9 and developing processes are carried out in regular order of the yellow developing device 20Y, the magenta developing device 20M, the cyan developing device 20C and the black developing device 21B. The intermediate transfer body 9 rotates four times so that the visible images composed of respective toners of yellow, magenta, cyan and black are sequentially formed. As a result, a full color visible image is formed on the intermediate transfer body 9.

The intermediate transfer body 9 is adapted to come into contact with the photosensitive drum 15 and rotate together with the photosensitive drum 15, and rotates clockwise upon formation of a color image and receives the multiple transfer of the visible image four times from the photosensitive drum 15. Further, during the image formation, the transfer roller 10 comes into contact with the intermediate transfer body 9 so that the intermediate transfer body 9 simultaneously performs the multiple transfer of the color visible image there onto the transfer material 2. On the outer periphery of the intermediate transfer body, are arranged a TOP sensor 9a and an RS sensor 9b for detecting positions concerning the rotating direction of the intermediate transfer body 9 and a density sensor 9c for detecting the density of a toner image transferred to the intermediate transfer body 9.

09699389-103100

The transfer roller 10 is provided with a transfer charger supported so as to be engaged with and disengaged from the photosensitive drum 15 and is configured by winding an intermediate resistance foaming elastic body on a metal shaft.

While the multiple transfer of the color visible image is carried out on the intermediate transfer body 9, the transfer roller 10 is, as shown by a solid line in Fig. 4, separated below from the body 9 so as not to disturb the color visible image. Then, after the four color visible image is formed on the intermediate transfer body 9, the transfer roller 10 is located at an upper position shown by a dotted line in Fig. 4 by a cam member (not shown) to a timing for transferring the color visible image to the transfer material 2. Thus, the transfer roller 10 comes into contact with the intermediate transfer body 9 under prescribed pressure through the transfer material 2 and applies a bias voltage to the intermediate transfer body 9 to transfer the color visible image on the intermediate transfer body 9 to the transfer material 2.

The fixing part 25 serves to fix the transferred color visible image to the transfer material while conveying the transfer material 2. The fixing part 25 is provided with a fixing roller 26 for heating the transfer material 2 and a pressing roller 27 for allowing the transfer material 2 to come into contact

00609389-103100

5

10

15

25

fixing part 25 to fix the toner image thereto. In the case of a single side printing, a guide 38 forms a conveying passage to guide the recording sheet to the upper sheet discharging part. As compared with the above, in the case of a duplex side printing, the guide 38 forms a path to guide the recording sheet to a lower duplex side unit.

The recording sheet guided to the duplex side unit is temporarily fed to the lower part (conveying passage shown by a two-dot chain line) of the tray 1 by a conveying roller 40, and then conveyed in a reverse direction and fed to a duplex side tray 39. On the duplex side tray 39, the front surface and the back surface of the sheet are reverse to those of the sheet which is mounted on the sheet feeding tray 1 and the head and the last of the sheet in a conveying direction are reverse to those of the sheet mounted on the sheet feeding tray 1. The toner image is transferred and fixed again to the sheet under this condition, so that the duplex side printing can be realized.

Fig. 5 shows a flowchart showing the processes of storing steps for each page in the generation of the spool file 303 in the spooler 302.

Initially, in step 501, the spooler 302 receives a print request through the graphic engine 202 from the application. In the application, a dialogue for inputting a print setting as shown in Fig. 8 is

09699389-103100
DOTED-68E6960

5

10

15

20

25

is for change of page. If the spooler 302 decides that the request is for change of page in step 507, the spooler advances to step 508 to inform the spool file manager 304 of the progress of the printing process.

5 Then, the spooler 304 increments a page number counter to close the page description file in which the intermediate code is stored and prepare a next page description file.

10 In step 507, when the spooler 304 decides that the received print request is not for change of page, the spooler 302 moves to step 509 to prepare to write the intermediate code on the page description file.

15 Subsequently, in step 510, in order to store the print request in the spool file 303, the spooler 302 converts the DDI function of the print request into the intermediate code. In step 511, the spooler 302 writes the print request (intermediate code) converted into a form which can be stored in the spool file in step 510 in the page description file of the spool file 303.

20 After that, the spooler 302 returns to the step 501 to receive again a print request from the application. A series of processes from the step 501 to the step 511 are continued until a job end request (End Doc) is received from the application. Further, the spooler
25 302 simultaneously obtains the information such as the process setting stored in the DEVMODE structure body from the printer driver 203 and stores the information

00609389-103100

5

10

15

20

25

logical page which is sent from the spooler 302
informed in the above described step 508 to the spool
file manager 304. Here, when the progress notice is
the print end notice of one logical page, the spool
5 file manager 304 moves to step 605 to store logical
page information corresponding to the logical page.
Then, in the next step 606, the spool file manager 304
decides whether or not the print operation of one
physical page can be started relative to n logical
10 pages in which a spooling process is finished at that
time. Here when the physical page can be printed, the
spool file manager 304 advances to step 607 to
determine a physical page number on the basis of a
logical value allocated to one physical page to be
15 printed.

As for the calculation of physical pages, for
instance, in the case where the print process setting
or the print modification setting is such a setting as
to arrange four logical pages for one physical page,
20 the first physical page can be printed when a fourth
logical page is spooled to obtain the first physical
page. Subsequently, a second physical page can be
printed when a eighth logical page is spooled.

Further, even when the total number of logical
25 pages is not a multiple of the number of logical pages
arranged on one physical page, the logical pages to be
arranged on one physical page can be determined in

001E01" 68E66960

accordance with a spool end notice in step 512.

Then, in step 608, information such as a logical page number, the physical page number thereof, etc.

which constitute a physical page to be printed is

5 stored in the setting file for outputting a job (a file including physical page information) in a form shown in Fig. 10 and the despooler 305 is informed of the addition of physical page information corresponding to one physical page. After that, the spool file manager 10 304 returns to step 601 to wait for a next notice. In the present embodiment, when the logical pages which constitute one page of printing data, that is to say, one physical page are spooled, the printing process can be executed even when the spool process of the print 15 job is not completely finished.

On the other hand, in step 604, when the progress notice is not the print end notice of one logical page from the spooler 302, the spool file manager 304 advances to step 609 to decide whether or not the 20 progress notice is the job end notice from the spooler 302 which is sent to the spool file manager 304 from the spooler in step 512. Here, when a replay is affirmative, the spool file manager 304 moves to the above described step 606. On the other hand, when a 25 reply is negative, the spool file manager 304 proceeds to step 610 to decide whether or not the received notice is a print end notice of one physical page from

00699389-103100
007001-68E66960

the despooler 305. Here, when the received notice is the print end notice of one physical page, the spool file manager 304 proceeds to step 612 to decide whether or not the print of the modification setting is completely finished. When the printing process is finished, the spool file manager 304 advances to step 612 to inform the despooler 305 of the end of printing. On the other hand, when the spool file manager 304 decides that the print of the modification setting is not finished, the spool file manager moves to step 606. In the despooler 305 according to the present embodiment, it is assumed that the number of one physical page is a unit for executing the print process. Further, in step 608, information required for performing the print process of one physical page is stored one by one in a file so as to be reused. When the information does not need to be reused, a package system, may be provided, in which a high speed medium such as a common memory is used to successively overwrite information thereon for each physical page so that speed and resources can be saved. Further, in the case where the progress of spooling is faster than that of despooling or in the case where a despooling process is started after all pages are spooled, the spool file manager 304 does not inform the despooler 305 of the pages which can be printed for each physical page in step 608, and can save the number of notices by

informing the despooler of the contents of a notice that a plurality of physical pages or all physical pages can be printed in accordance with the progress of the despooling side.

5 In step 610, when the spool file manager 304 decides that the received notice is not the print end notice of one physical page from the despooler 305, the spool file manager 304 advances to step 613 to decide whether or not the notice is a print end notice from 10 the despooler 305. When it is decided that the notice is the print end notice from the despooler 305, the spool file manager 304 advances to step 614 to delete a corresponding description file of the spool file 303 and finish the processes. On the other hand, when the 15 notice is not the print end notice from the despooler 305, the spool file manager 304 proceeds to step 615 to perform other ordinary processes and wait for a next notice.

Fig. 7 is a flowchart showing the detail of the 20 print data generating process in the despooler 305.

The despooler 305 reads out necessary information (page description files and job setting files) from the spool file 303 in accordance with a print request from the spool file manager 304 to generating printing data. 25 The generated printing data is transferred to the printer in accordance with the method explained with reference to Fig. 3.

00559389-103100

For generating the printing data, in step 701, the despooler 305 first inputs a notice from the spool file manager 304. In next step 702, the despooler 305 decides whether or not the inputted notice is the job end notice. When the inputted notice is the job end notice, the despooler 305 advances to step 703 to raise an end flag and move to step 705. On the other hand, in step 702, when the notice is not the job end notice, the despooler 305 advances to step 704 to decide whether or not the print start request of one physical page in step 608 is notified to the despooler. In step 704, when the despooler 305 decides that the notice is not the print start request, the despooler 305 advances to step 710 to carry out other error processes, return to step 701 and wait for a next notice. On the other hand, in step 704, when the despooler 305 decides that the notice is the print start request of one physical page, the despooler 305 advances to step 705 to store the ID of the physical page to be printed which the despooler receives the notice in step 704. In the next step 706, the despooler 305 decides whether or not the print processes of all the pages of the ID of the physical page stored in step 705 are completed. Here, when the processes of all the physical pages are completed, the despooler 305 advances to step 707 to decide whether or not the end flag in step 703 is raised. When the end flag is raised, the despooler 305

00699389 103100
007E07 68E66960

decides that the print process of the job is finished and sends a notice of the end of processes to the spool file manager 304 to finish the processes. In step 707, when it is decided that the end flag is not raised, the despooler 305 returns to the step 701 to wait for a next notice. On the other hand, in step 706, when the despooler 305 decides that there remain physical pages to be printed, the despooler advances to step 708, sequentially read IDs of physical pages to be processed from the stored physical page IDs, read information necessary for generating the printing data of the physical pages corresponding to the read physical page IDs and execute the print process. The print process is performed by converting a print request instruction stored in the spool file 303 into a format (GDI function) which can be recognized by the graphic engine 202 in the despooler 305 and transferring the format. As for the process setting or the modification setting (called it N page printing, hereinafter) in which a plurality of logical pages are laid out on one physical page. The conversion is executed by considering a reduced arrangement in step 708. When the required print process is finished, the despooler 305 notifies the spool file manager 304 of the end of print data generation of one physical page in next step 709. Then, the despooler 305 returns to the step 706 to repeat the processes up to the print process for all

5

10

15

20

25

In the print process using the spooler 302, when the print request is supplied to the graphic engine 202, the job output setting file is generated by the despooler 305. In the case where the previews or the jobs are combined together, the job output setting file is also generated. The job output setting file is, in

5

25

to be printed, this field may not be required for operation. The physical page information of the number shown in the field 1003 is stored from a field 1004 to the last field. The physical page information will be described by referring to Fig. 12.

Fig. 11 shows one example of the job setting information shown in the field 1002 shown in Fig. 10. A field 1101 designates the number of all physical pages. A field 1102 designates the number of all logical pages. The fields 1101 and 1102 are added to the printing data and employed when a page number is printed as additional information. While the printing operation is continued, both the fields designate provisional values, or the spool manager 304 postpones the preparation of physical page information capable of being printed until the printing operation is finished. A field 1103 designates copy number information for designating what number of copies of the print job are printed. A field 1104 designates whether or not the print job is printed for each copy when a plurality of copies are printed in the field 1103. A field 1104 designates finishing information such as stapling, punching, folding in a Z-shape, etc. When a finisher is provided externally or in the printer main body, the finishing information is designated. A field 1106 indicates additional print information, and a decoration such as a page frame, additional information

001E01" 68E66960

such as date, and information added to a job such as a user name, a page number, water mark print, etc. are stored. As functions increase, the number of fields included in the job setting information also increases. For example, when a duplex side printing can be carried out, a field for storing the designation of the duplex side printing is added thereto.

Fig. 12 shows one example of the physical page information shown in the field 1004 in Fig. 10. A first field 1201 denotes physical page numbers which are values used when a print order is controlled and a physical page number is added and printed. A field 1202 denotes physical page setting information in which the setting of layout or color/monochrome is stored when the layout or the color/monochrome can be designated for each physical page. A field 1203 designates the number of logical pages allocated to the physical page. If four pages are allocated to one physical page, an ID indicating 4 or a four page printing will be stored. After a field 1204, the information of logical pages of the number designated by the field 1203 is stored. The actual number of page data may be possibly smaller than the number of pages designated by the field 1203 depending on the number of pages printed from the application 201. In order to meet such a case, special data showing a space page is stored in the logical page information.

00509389-103100

Fig. 13 shows an example of the physical page setting information indicated by 1202. A field 1301 denotes the arrangement order of logical pages on a physical page in which the designation of an order (transversely from an upper part in a left side, downward from an upper part in a left side, etc.) for arranging the logical pages on a physical page on the basis of the N page printing is stored. The arrangement order may not be applied depending on a system. The logical page information after the field 1204 may be arranged in order in accordance with the arrangement order without using a page number sequence in place of the setting of 1301. A field 1302 indicates the obverse/reverse side information of the duplex side printing which is employed, for example, upon registration of a binding margin in the obverse/reverse side. A field 1303 denotes the designation of color page/monochrome page. In the case of a printer having a monochrome mode and a color mode, the above designation is a value used when the user wishes to print color pages by the color mode and monochrome pages by the monochrome mode in a document with the color pages mixed with the monochrome pages. The provision of this information enables the processes to be changed for each page as an auto-color mode in a color printer. Specifically, in the case of the color page, an intermediate transfer body (an intermediate

5
10

15

20

25

5

15

25

5

10

15

20

25

00609389-103100

a change between the contents of the setting first read in step 1501 between the contents of the setting designated at present. When the contents of the setting are changed, the setting change editor 307

5 advances to step 1505 to generate a new job output setting file, inform the spool file manager of the presence of a change and finish the processes. On the other hand, when the setting change editor decides that the contents of the setting are not changed in step

10 1505, the setting change editor sends a notice of no change to the spool file manager to finish the processes. As described above, the setting change editor 307 generates the new job output setting file. When an "OK" button is selected on the user interface

15 screen shown in Fig. 18, the new job output setting file becomes effective and the old job output setting file is deleted. Further, upon change in the job setting file of a single job different from that of the job output setting file, the job setting file is not

20 deleted but stored. Further, when a "return to initial state" button is selected on the screen of Fig. 18, the new job output setting file is deleted and the old job output setting file becomes effective to be displayed. In the present embodiment, although the setting change

25 editor 307 is described as a separate module, it may simply form a part of the user interface of the spool file manager 304. A package form, may be provided, in

which the setting change editor 307 does not actually write the contents of change in the job output setting file and only informs the spool file manager 304 of the contents of change of the setting so that the spool
5 file manager 304 side actually changes the job output setting file.

Fig. 3 further shows the extending system for combining a plurality of print jobs together and printing them as one print job. Now, an extending
10 processing for despooling and previewing the composed job will be described below.

Generally, the spool file 303 of the intermediate format is prepared for each job. In the case of a single job, the intermediate codes of each logical page
15 in a job file to be processed are sequentially read out to execute a process, so that the logical page ID of the field 1401 can be realized by a relative or absolute offset showing where each logical page is located in the file. In the case of the composed job,
20 it is necessary to specify the spool file and the page information belonging to the job from the job ID of the field 1401. In the present embodiment, there is utilized a system in which the ID for identifying the spool file is added to a logical page ID to specify the
25 spool file. In this case, a main change may be located only in the field 1401. If the spool file can be identified, the page part can be read in the same logic

00699339-103100

as that of the processing of the single job. Further,
in the case where the spool file is stored for each
logical page in a form of another file, there may be
provided a package form that the file name of the
5 logical page is directly identified as the logical page
ID of the field 1401.

[Combining Process Flow]

Now, the operation flow of the user will be
described below when a plurality of documents are
10 combined together. Fig. 19 shows a flowchart for
explaining the user operation until the documents are
listed up on the screen of the spool file manager 304.

In step 1901, upon start of the printing operation
on the application, the spool file manager 304 opens
15 the dialog of the printer driver to designate the
dialog to be spooled to the intermediate file format
operable on the user interface. In the example of the
dialog of the printer driver shown in Fig. 9, the
designation of "store" is selected from a combo box
20 901. Then, the spool file manager 304 advances to step
1902 to start the print process (this is not a true
print process and the application sees as if the actual
print process were executed, in other words, a
background print process is carried out and the
25 application outputs a description instruction to the
graphic engine), the print jobs prepared in the
intermediate data format are listed up on the screen of

00609389-103100

the user interface of the spool file manager 304 as shown in Fig. 16 so that the designation of the user can be received while the print jobs are spooled in the intermediate file format. Under this state, the user
5 can select and print one document from the list, temporarily change the print setting or recognize the contents of the setting on the screen.

In this case, since the user intends to combine a plurality of printed documents together, the spool file
10 manager 304 advances to step 1903 to set the "store" to all the documents to be combined together and decide whether or not the documents are printed. When there remain documents which are not printed (the output
15 process of the description instruction to the graphic engine) if the documents to be combined together, the spool file manager 304 returns to the step 1901 to repeat the print processes of other documents to be combined together. When the printing process of all
20 the documents to be combined together are completed, the spool file manager advanced to a combining process.

Fig. 20 shows a flowchart for explaining the user operation upon combining together a plurality of documents listed up in Fig. 16.

In step 2001, the user allows to select desired
25 documents to be combined together on the user interface shown in Fig. 16 by using a pointing device such as a mouse. When the print jobs in the intermediate data

006090300-103100

5

10

15

20

25

embodiment, when the contents of one setting which are to be necessarily set to one print job is designated, a control is always executed so that a setting cancel message is displayed, however, the present invention is not limited thereto. It is to be understood that a control may be performed so that the setting cancel message is displayed when the contents of the settings of a plurality of print jobs to be combined together are examined and the contents of the settings which can be respectively necessarily set to one print job are different between a plurality of print jobs.

In next step 2003, the spool file manager 304 controls to display the user interface (see Fig. 23) on which the user can recognize the contents of the message and select as to whether the print jobs are combined or not and decides whether or not the user instructs the print jobs to be combined.

In step 2003, when the spool file manager 304 decides that the user instructs not to combine the print jobs together, the spool file manager returns to the step 2001 to select the document (print job) again. The spool file manager 304 may further return to the printing flow shown in Fig. 19 from the step 2001 to print a new document or to change the setting of the already printed document and print the document thus obtained again.

In step 2003, when the spool file manager 304

00509389.103100

5

10

15

25

printed in accordance with the setting of two copies and one page printing.

In the case where the document A is combined with the document B, the number of copies is reset to one, the part of the document A is set to a four-page printing and the layout information of the document B has the setting of one page printing. As for the number of copies, the setting on a unit for processing the print job (the designation in Fig. 11), that is to say, one setting can be necessarily set to one print job, so that the composed job needs to be unified or reset to a new value. The page layout (layout information) includes the settings to the individual documents and pages, in other words, information held by the physical page information (see 1203 in Fig. 12) and does not need to be unified so that it is printed while the settings of the original documents are maintained.

A reason why the settings are changed or not changed in such a way upon combining the documents will be described below. Generally, there exist a number of kinds of print settings. They are classified into only one print setting which can exist for all the job obtained after the jobs are combined together and the printing settings which can exist in the respective jobs of the composed job and for each page. These print settings are shown in Fig. 26.

00100 103100 6866960

5
10
15

20

25

20 Next, an internal data structure in the combining
process will be described. Fig. 27 shows one example
of a data structure (job management list) for managing
the jobs listed up on the spool file manager 304. The
spool file manager 304 adds the ID of the intermediate
25 file to the job management list every time the
intermediate file is spooled. As for the ID of the
intermediate file, information which can specify the

intermediate file from the ID, for example, are used as the ID a file name if the intermediate file is located in a file system, a pointer if the intermediate file is present in a memory, and so on.

5 Fig. 28 shows one example of the job management
list when the spool file manager 304 manages the
composed jobs. In this example, in order to
discriminate the composed job from ordinary job, a pair
of a job ID and a flag for indicating whether or not
10 the job is a composed job represents one job. When the
flag shows the ordinary or regular job, the ID of the
intermediate file is stored. When the flag shows the
composed job, the ID of the composed job is stored.

In this case, since the composed job is managed in the memory, a pointer to a structure body which represents the composed job is stored. The structure body representing the composed job comprises the number of jobs in the composed job and the ID of the intermediate file of each print job in the composed job. In such a way, the intermediate file in which the data is actually stored is not directly operated but a logical value of the Job ID is operated, so that resources necessary for a job operation can be saved. Further, even when the jobs are repeatedly combined together, the structures of the files are maintained, and therefore, the jobs can be reset to their states when they are spooled.

Fig. 32 is a flowchart showing job operation processes in the spooler 302 and the spool file manager 304.

5 In step 3201, the spooler 302 or the spool file manager 304 first receives a notice or print start notice from the user.

10 In step 3202, the spooler 302 decides whether or not the received notice is the job print notice of a store setting. In the step 3202, when the spool file manager 304 decides that the received notice is the print start notice of the store setting, the spool file manager 304 advances to step 3203 to register the intermediate file ID of the noticed print job in the job management list held by itself and return to the
15 step 3201 in order to receive a next notice.

20 In the step 3202, when the spool file manager 304 decides that the received notice is not the job print start notice of the store setting, the spool file manager advances to step 3204 to decide whether or not the received notice is an instruction notice for combining the jobs together illustrated in Fig. 16. When the received notice is the instruction notice for joining the jobs together, the spool file manager 304 advances to step 3205 to prepare a composed job
25 structure for managing the composed jobs by using the setting change editor 307.

Then, the spool file manager 304 advances to step

001E01" 68E66960

3206 to copy all the intermediate file IDs of the jobs to be combined together which are currently selected to the composed job structure prepared in the step 3205.

Then, the spool file manager 304 advances to step 3207

5 to delete the jobs copied to the composed job structure from the job management list held by itself and add a composed job structure newly prepared in the step 3205 to the job management list. Then, the spool file

10 manager 304 returns to the step 3201 to wait for a next input.

In the step 3204, when the received notice is not an instruction for combination of the jobs, the spool file manager 304 further advances to step 3208 to

15 decide whether or not the received notice is an instruction to delete the jobs. When the notice is the instruction to delete the jobs, the spool file manager 304 advances to step 3209 to decide whether a job instructed to be deleted is a composed job or an ordinary job. Here, when the spool file manager

20 decides that the job is the composed job, the spool file manager 304 advances to step 3210 to delete the composed job structure located at the end of the pointer which is registered in the job management list, and moves to step 3211 to delete the job instructed to
25 be deleted from the job management list.

In the step 3209, when the spool file manager 304 decides that the job instructed to be deleted is not a

09699389 103100

composed job, the spool file manager 304 jumps step
3210 and advances to step 3211 to delete the job from
the job management list. Then, the spool file manager
304 returns again to the step 3201 to wait for a next
5 input.

10 In the step 3208, when the spool file manager 304
decides that the received notice is not the instruction
to delete the jobs, the spool file manager 304 proceeds
to step 3212 to decide whether or not the received
notice is an instruction to duplicate the job. When
the spool file manager 304 decides that the notice is
the instruction to duplicate the job, the spool file
manager 304 adds a job ID to be duplicated which is
currently selected to the job management list. In the
15 case of the composed job, a pointer showing the
composed job structure may be similarly copied to the
job management structure. However, according to the
simply copying system, a management upon deletion of
the job is complicated, so that the number of times of
20 referring to the job needs to be stored in another
list. Otherwise, may be adopted a management system
the intermediate file or the composed job structure
itself is copied and another ID is attached thereto.
In this case, a deletion process can be simply managed,
25 however, time required for copying the intermediate
file or resources for storing the copied intermediate
file are excessively consumed, these systems need to be

001001" 6866960

Then, the spool file manager 304 returns again to the step 3201 to wait for a next input. In the step

5 3212, the spool file manager 304 decides that the
received notice is not the instruction to duplicate the
job, the spool file manager 304 advances to step 3214
to decide whether or not the notice is an instruction
to decompose the job. When the spool file manager 304
10 decides that the notice is the instruction to decompose
the job, the spool file manager 304 proceeds to step
3215 to return all the job IDs held in the composed job
structure indicated by the job pointer of the job to be
decomposed to the job management list, delete the job
15 to be decomposed from the job management list and also
delete the composed job structure. Then, the spool
file manager 304 displays the job icon of the single
job on the user interface screen illustrated in Fig.
16.

20 Further, before moving to step 3215, a message for recognizing whether the composed job is to be truly decomposed may be displayed as illustrated in Fig. 25. Then, the spool file manager 304 returns again to the step 3201 to wait for a next input.

25 Now, the operation of the composed job will be
described below. The data structure used upon preview
of the print process of the composed job is managed by

5

10

15

20

25

according to the present system which cannot be
achieved by the prior art, a job 1, a job 3 and a job 5
are successively spooled by the application A, a job 2
is spooled by the next application B, and a job 4 is
5 sequentially spooled by the application C, the jobs are
combined together, and then, the order change buttons
are operated, so that they can be arranged in regular
order such as the job 1, the job 2, the job 3, the job
4 and the job 5 and an operability is extremely
10 improved. With the order change, physical page numbers
(or job IDs) of the job output setting file are
updated.

Further, a job deleting interface designated by
3003 is operated so that the print job can be deleted
15 from the composed job. The print job is deleted so
that a corresponding physical page number or logical
page number in the job output setting file is updated.

Fig. 33 is a flowchart showing a processing flow
when jobs are reshuffled and jobs are deleted.

20 In step 3301, the spool file manager 304 receives
an input from the user.

Then, the spool file manager 304 advances to step
3302 to decide whether or not the input from the user
is a request to delete a job. When the spool file
25 manager 304 decides that the input from the user is the
deletion request for the job, the spool file manager
304 advances to step 3303 to decide whether or not the

00600330-103100
00TET" 68E66960

5 input without performing any process. In the case where data is not present in the composed job after the deletion of the job in accordance with the process of the user interface, it is to be understood that the user may not select the deletion from the first.

In the step 3302, when the spool file manager 304
25 decides that the input from the user is not the request
to delete the job, the spool file manager 304 advances
to step 3305 to decide whether or not the input from

the user is an instruction to rearrange the job order.
In the step 3305, when the spool file manager 304
decides that the input from the user is the instruction
to rearrange the job order, the spool file manager
5 advances to step 3306 to replace the ID of the job to
be moved which is currently selected by the ID of a job
to which the former moves. Fig. 31A shows an example
in which the second job is moved to the first position
in accordance with a change of the data structure upon
10 change of the job order. The job ID held in the
position of the first job of the composed job is
replaced by the job ID held in the position of the
second job, so that the order change of the jobs can be
realized. In the step 3305, when the spool file
15 manager decides that the input from the user is not the
instruction to rearrange the job order, the spool file
manager 304 performs a process in accordance with other
instructions to return to the step 3301.

Now, an operation for each page will be described
20 hereinafter. The page can be deleted for each logical
page in accordance with a control process indicated by
3004 in Fig. 30. In this case, the physical page
information file is operated. Since the preview is
displayed in 3005 of Fig. 30, the physical page
25 information file is already prepared from the job
management list.

Fig. 34 is a flowchart for explaining a process

00699389-103100
001E01-68E66960

5

10

15

25

5 or not the first page of the subsequent job B is
printed on the back side of the sheet, any print is not
performed on the back side of the last page so that the
first page is printed on the front surface of a next
sheet depending on the setting, the first page of the
10 job B is successively printed on the back surface, or
the physical page information file is configured to be
divided.

In the step 3406, when the spool file manager 304 decides that the page is located in the job boundary, the spool file manager 304 advances to step 3407 to perform the process of the job boundary and moves to step 3408. In the step 3406, the spool file manager decides that the page is not a job boundary page, the spool file manager 304 advances to the step 3408. In the step 3408, the spool file manager 304 shifts a page subsequent to the logical page to be processed so as to fill an unoccupied page with the page. Then, the spool file manager 304 advances to step 3409 to move backward the logical page to be processed by one and returns to the step 3403. As a result of the successive repetitions, when the processes advance to the last page of the composed job, the spool file manager 304

In the step 3404, the spool file manager 304 decides whether there is no logical page on the final physical page to be currently processed as a result of moving the page. When there is not any logical page to be displayed on the physical page, the spool file manger 304 advances to step 3405 to delete the physical page information itself. When the physical page is empty and the logical page does not exist, the spool file manager 304 finishes the process without deleting the physical page.

As described above, according to the present embodiment, the information processor for generating printing data to be transmitted to a printer comprises an intermediate code converter for converting data to be printed which is generated by an application into the print jobs of intermediate code format and temporarily storing the print jobs in association with the print setting information of the data to be printed; a composition instructing unit for instructing a plurality of print jobs corresponding to the different data to be printed to be combined together so as to generate one composed job; and a setting unifier for analyzing the print setting information of a plurality of print jobs when the composition instructing unit instructs the plurality of print jobs

to be combined together so as to obtain one composed job, and generating print setting information for the composed job in which information that can be respectively merely set to one print job is unified.

5 Therefore, a plurality of print jobs respectively having the print setting information can be joined together. Further, the inconvenience in the print setting upon joining the print jobs together can be eliminated.

10 Further, since the setting unifier further includes a recognizing unit for analyzing the print setting information of a plurality of print jobs to be combined together and recognizing to select whether the settings are unified or the print jobs are not combined
15 together when information which can be respectively set only to one print job is mutually different, the combination or composition of the jobs which is not desired by a user can be avoided.

[Layout Unifying Process]

20 Figs. 35 to 39 respectively roughly show a processing flow for explaining a " layout unifying process" in the spool file manager 304 and the setting change editor 307 which is one feature of the present invention. Figs. 40 to 41 respectively show examples
25 of the setting change editor 307. The summary of the present invention will be described by using the above described means.

00100 68E66960

As described above, when a plurality of print jobs are stored on the spool file manager 304, the list of the print jobs are displayed as illustrated in Fig. 16 so that the spool file manager 304 waits for an operation instruction. Then, when a plurality of jobs are selected from the list and a combining instruction set to the menu or the icon is supplied, it is decided in a deciding step S3501 for deciding whether or the combining instruction is supplied that the combining instruction is supplied. Then, the setting change editor 307 as shown in Fig. 18 is started in the start step S3502 of the setting change editor 307. Then, as described above, the job output setting files (see Figs. 10 to 14) are generated. Thus, while the print jobs spooled in the intermediate files are listed up and displayed as shown in Fig. 16, when a plurality of print jobs are selected and the instruction of "combination of jobs" is supplied, the instructed print jobs are simply sequentially combined together from an upper part. In other words, the page layout information for each physical page of the composed job is prepared in the job output setting file on the basis of the job setting files (including the page layout information) of the respective single jobs.

Then, in order to display a print setting tab and the preview shown in Fig. 18, a next initial layout setting obtaining step S3503 is performed. Here, the

5
10

15
20
25

employed as they are upon combining the jobs together.

Then, the state of the initial layout is displayed on the print setting tab in Fig. 18 by a layout setting display step S3504. As described above, since the layout in which the respective print jobs are held as they are is used (not the layout unified by the prescribed rule), the arrangement order of the layout or the like is displayed in grey. Then, a preparing step S3505 of the physical page information file is carried out. This step is a process for preparing the physical page information files shown in Figs. 10 to 14 on the basis of the determined layout. This process will be described below in more detail. A next preview display step S3506 is performed. This is a process for displaying the preview part shown in Fig. 18. Here, since the layout in which the respective print jobs are held as they are is employed, the composed job is formed and displayed while the layout before a plurality of print jobs are combined together as illustrated in Fig. 40 is maintained. More specifically, in Fig. 40, a job 1 corresponds to one logical page of "one page/sheet", a job 2 corresponds to four logical pages of "four pages/sheet", and a job 3 corresponds to one logical page of "one page/sheet". In the case where the composed job is formed by these three jobs, the layout information thereof are respectively individually stored, so that a small

00609389 103100

preview with a plurality of layouts mixed is obtained. Besides, even when a large preview is displayed, the preview in which a plurality of layouts are mixed likewise is displayed.

5 Now, for preparing the layout, the detail of the preparing step S3505 of the physical page information file will be described by way of Figs. 37 and 38.

10 Initially, the calculating process 3701 of the number of all the physical pages is carried out. This is a process for calculating how many physical pages the composed job has. Here, since the layouts of the jobs before joined are held, the number of jobs is equal to the number of all the physical pages of the jobs to be combined together. Then, the calculating
15 process 3702 of the physical page information is performed. This is a process for preparing the physical page information of each job which constitutes the composed job. The detail of the process is shown in Fig. 38.

20 In the calculating process of the physical page information, a calculating process 3801 (see Fig. 38) for calculating the number of the logical pages allocated to the physical page is first executed. This is a process for calculating the number of logical
25 pages allocated to each physical page. In the case of the first physical page shown in Fig. 40, 1 equal to the number of jobs before combined together is

00690389.103100
007E07" 68E66960

5
10
15
20

25

deciding whether the printing setting is the single
side printing or the duplex side printing. When it is
decided that the print setting is the duplex side
printing, a next obverse/reverse side writing process
5 3902 is carried out. This is a process for deciding
whether the physical page is located in the obverse
side or the reverse side and writing the
obverse/reverse side data. Here, the single side
printing is supposed to be executed, so that the data
10 is not written. Then, a deciding process 3903 for
deciding whether or not the print setting is a color
printing is performed. This is a process for deciding
whether the print setting is the color printing or a
monochrome printing. Here, when it is decided that the
15 print setting is the color printing, a next
color/monochrome writing process 3904 is carried out.
This is a process for deciding whether or not the
logical pages to be printed in color are included in
the physical page. When the logical pages to be
20 printed in color are included, color data is written,
and when the logical pages are not included, monochrome
data is written. Here, the monochrome printing is
supposed to be executed, so that the monochrome data is
not written. In such a way, the physical page setting
25 information is generated and obtained.

Then, the processes return to those of Fig. 37 to
calculate the physical page information corresponding

006090300-103100
00FE0T" 50E66960

5 The physical page information file is prepared and
the layout is formed in accordance with a series of the
above described processes.

10

15

25

instruction to unify the layouts. Then, an obtaining process 3602 for setting a page layout, an arrangement order, and an intrusion. The initialization thereof is obtained in accordance with a predetermined rule. This rule may include various kinds of rules. For instance, the layout is adapted to meet the layout of the initially selected print job on the spool file manager 304, to meet that of the first print job on the list, to meet most of the layouts of a plurality of print jobs, to meet the layout predetermined as the setting of a default, etc. Here, it is assumed that the layout is adapted to meet the layout predetermined as the setting of a default, and, in the layout of 4up, from an upper part in the left side to the right, the intrusion of the jobs is determined to be permitted. As for the intrusion of the jobs (page shifting process), for instance, in the case where the number of logical pages is 3 in the layout of 4-up, the logical page equivalent to one remaining page can be compactly shifted. In this case, can be performed a selection for deciding whether the first logical page of a next job is laid out on the last of the physical page (intrusion designation), the first logical page is laid out on the first of another physical page (designation without intrusion) or the first logical page is laid out on the first of the back side (intrusion designation on reverse side).

00609389-103100

Then, an editing process 3603 of the physical page information file is carried out to the setting mentioned above. This is a process for editing the physical page information files shown in Figs. 10 to 14 according to the determined layout. This process will be described later in more detail. A next preview process 3604 is carried out on the basis of the physical page information files edited in the above process. This is a process for displaying the preview part shown in fig. 18. Here, since the layout of 4 up from the upper part in the left side to the right in which the intrusion of the jobs is permitted is adopted, the composed job is formed and displayed as illustrated in Fig. 41. On the job output setting file, "4" is designated as the number n of the logical pages allocated to all the physical pages of the physical page information (see 1203 of Fig. 12).

Now, in connection with the formation of the layout, the detail of the editing process 3603 of the physical page information file concerning the job output setting file will be described below.

The calculating process 3701 for calculating the number of all the physical pages. This is a process for calculating again how many physical pages the composed job has in accordance with the designation of "unify layout" of the composed job. If the setting is such a setting as to permit the intrusion of the jobs,

00690389-103100

the number of physical pages will be changed.
Considering the examples of figs. 40 to 41, 2 physical
pages are obtained. Then, the calculating process 3702
for calculating the physical page information is
5 carried out. This is a process for preparing each
physical page information forming the composed job and
the detail thereof is shown in Fig. 22.

In the calculating process 3702 of the physical
page information, the calculating process 3801 (see
10 fig. 38) for calculating the number of logical pages
allocated to the physical page is first carried out.
This is a process for calculating the number of logical
pages allocated to each physical page. In the case of
the first physical page shown in Fig. 41, a prescribed
15 value, or "4" is calculated in the present embodiment
by unifying the layouts. As described above, the
number of logical pages allocated to the physical page
may be determined on the basis of the layout
information of the first single job of the composed
20 job. In this case, since the job 1 is designated by
"1", the number of logical pages is "1".

Then, a calculating process 3802 of the logical
page information is carried out. This is a process for
calculating each logical page information indicating
25 the physical page which is a process for calculating
the information as shown in fig. 14. Then, a deciding
process 3803 for deciding whether or not the logical

00609389 103100
001E01 58E6960

page is the last and an increment process 3804 of a logical page counter are carried out, so that the logical page information corresponding to the number of logical pages which is previously calculated is
5 obtained. And the number of the logical pages is obtained. Then, a calculating process 2205 of the physical page setting information is performed. This is a process for calculating the physical page setting information shown in fig. 13. The detail of the
10 calculating process is shown in Fig. 39.

In the obtaining process 3805 of the physical page setting information, a deciding process 3901 for deciding whether the duplex side printing is performed is first carried out. This is a process for deciding
15 whether the printing setting is the single side printing or the duplex side printing. When it is decided that the print setting is the duplex side printing, a next obverse/reverse side writing process 3902 is carried out. This is a process for deciding
20 whether the physical page is located in the obverse side or the reverse side and writing the obverse/reverse side data. In the case of the duplex side printing, the logical pages allocated to the obverse side and the reverse side may be possibly
25 changed due to the unification of the layouts. Here, it is assumed that the single side printing is employed and the data is not written.

00609389-103100

Then, a deciding process 3903 for deciding whether or not the print setting is a color printing is performed. This is a process for deciding whether the print setting is the color printing or a monochrome printing. Here, when it is decided that the print setting is the color printing, a next color/monochrome writing process 3904 is carried out. This is a process for deciding whether or not the logical pages to be printed in color are included in the physical page.

When the logical pages to be printed in color are included, color data is written, and when the logical pages are not included, monochrome data is written. Also in this case, the color pages included in the physical page may be possible changed by unification of the layout. Here, the monochrome printing is supposed to be executed, so that the monochrome data is not written.

In such a way, the physical page setting information is generated and obtained. The job output setting file including the physical page setting information thus generated is formed by copying the original job output setting file before the layouts are unified and writing the physical page setting information in the job output setting file thus formed. Thus, when a "return to initial state" button is pushed down on the user interface for editing the jobs (shown in Fig. 18 or Fig. 40), the contents of the job output

00699389-103100

5 duplicated and updated job output setting file is
overwritten on the original job output setting file, so
that the contents of the job output setting file before
the layouts are unified are erased. In this case, in
the user interface shown in Fig. 16, when the user
10 selects the composed job and instructs the composed job
to be decomposed, the composed job is decomposed to the
individual single jobs. The respective job setting
files are always held in the spool file 303 for the
single jobs. Therefore, the most primary print setting
15 is unchangeably ensured.

20 deciding whether or not the physical page is the last
or the increment process 3704 of the physical page
counter

25 is regenerated. It is to be understood that the present invention may be applied to a system comprising a plurality of devices such as a host computer,

As described in the above embodiment, since the information processor according to the present

Further, the objects of the present invention can be achieved by reading and executing a program code stored in a memory medium in which the program code of software for realizing the above described functions of the embodiment by the use of the computer or a CPU 101 or an MPU) of a system or an apparatus.

As the memory medium for supplying the program

code, may be utilized, for example, a floppy disk, a hard disk, an optical disk, a magneto-optic disk, a CD-ROM, a CD-R, a magnetic tape, non-volatile memory card, a DVD, an MO, a ROM, etc.

5 The program code read by the computer is executed, so that not only the above described functions of the embodiment can be achieved, but also a part or all of the actual processes are executed by the OS (operating system) operating on the computer on the basis of the
10 instruction of the program code or the like and accordingly, the above described functions of the embodiment can be achieved.

 Further, after the program code read from the memory medium is written in a memory provided in a
15 function extending board inserted into the computer or a function extending unit connected to the computer, the CPU or the like provided in the function extending board or the function extending unit performs a part or all of the actual processes on the basis of the
20 instruction of the program code, so that the above described functions of the embodiment can be achieved by the processes.

 Still further, the objects of the present invention can be also achieved by installing the
25 program code of the software for realizing the functions of the embodiment in the computer (or the CPU and the MPU) of the system of the apparatus and reading

00609389-103100

and executing the installed program code.

In this case, the program code itself downloaded in the download service of the internet and installed realizes the above mentioned functions of the embodiment and the program code itself constitutes the present invention.

Further, in order to the functional processes of the present invention, the program code itself installed in the computer also realizes the present invention. In other words, in the claims of the present invention, a computer program itself for realizing the functional processes of the present invention is also included. A method for supplying the computer program is not limited to a case in which the computer program is stored in the FD or the CD-ROM, read by the computer and installed in the computer. The computer is connected to the home page of the internet by using the browser of a client computer so that the computer program itself of the present invention, or a file compressed and having an automatic installing function is downloaded so that the computer program itself can be supplied. Further, the program code constituting the program of the present invention is divided into a plurality of files and respective files are downloaded from different home pages so that the computer program itself can be supplied. That is to say, a WWW server for downloading a program file for

00699389-103100

realizing the functional processes of the present invention by the computer to a plurality of users is also included in the claims of the present invention.

Still further, the program of the present invention is encoded, stored in the memory medium such as the FD and the encoded program is supplied to the users, then, key information for decoding the encoded program is downloaded from the home page through the internet by the user who clears prescribed conditions, and the key information is used by the user to execute the encoded program and install the program in the computer.

As many apparently widely different embodiments of the present invention can be made without departing from the spirit and scope thereof, it is to be understood that invention is not limited to the specific embodiments thereof except as defined in the appended claims.

As mentioned above, the information processor according to the present invention comprises an intermediate code converter for converting data to be printed which is generated by an application into print jobs of an intermediate code format and temporarily storing the print jobs in association with the print setting information of the data to be printed; a composition instructing unit for instructing a plurality of print jobs of the intermediate code format

00600389-103100

5 composition instructing unit instructs the plurality of
print jobs to be combined together so as to obtain one
composed job and generating print setting information
for the composed job in which information that can be
respectively merely set to one print job is unified.

Further, the above described setting unifier
15 serves to analyze the print setting information of a
plurality of print jobs to be combined together and the
information processor of the present invention further
includes a recognizing unit for recognizing to select
whether the settings are unified or the print jobs are
20 not combined together when the information which can be
respectively set only to one print job is mutually
different.

Still further, since information for designating

the page description files laid out on a physical page is added to the print setting information for the composed job, the description data of the intermediate data format does not need to be included in the composed job itself, the process upon combination of the jobs can be executed with ease.

Still further, since the setting information is temporarily stored as a print setting file of each print job, the job can be treated more easily.

Still further, since the information processor further comprises a preview display controller for controlling a preview based on the print setting information of the print jobs or the composed job to be displayed, the user can recognize a printed result before printing.

Still further, since the information processor further comprises an order controller for operating a plurality of print jobs in the composed job to reshuffle the order of the print jobs, the user can prepare the intermediate code without especially considering a print order upon preparation of the intermediate code and can determine the print order as desired.

Still further, since the information processor further comprises a job deleting unit for operating a plurality of print jobs in the composed job to delete a specific print job, it is not necessary to generate

09699389 "103100

again the composed job.

Still further, since the information processor according to claim 1, further comprises a job divider for dividing the composed job into a plurality of print jobs before they are joined together, the composed job
5 can be returned to respective single jobs. Accordingly, the user can easily try to generate the composed job.

Still further, since the information processor
10 further comprises a job duplicating unit for designating the print job or the composed job to prepare the duplication of the designated print job, a duplication can be prepared to control a page when the user desires to include another job in a job.

Still further, since the print job or the composed
15 job further includes a setting initializing unit for returning the data to an initial state upon preparation of the data of the intermediate code format as the base of the job on the basis of the print setting
20 information, the user can stably execute an operation such as editing the job.

Still further, since the information processor further comprises a page editing unit for deleting a page designated relative to a logical page in the print
25 job or the composed job, a job operation can be performed as desired by the user and an operability and a degree of freedom are improved.

00699389 103100
007E07 68E66960

Still further, since the information processor further comprises a printing data generator for generating the printing data to be transmitted to the printer on the basis of the data of the intermediate code format which is temporarily stored by the intermediate code converter, a desired job is generated and then, the generated job can be directly printed without restarting the application.

Still further, since the information processor further comprises a description instruction generator for converting the data of the intermediate code format temporarily stored by the intermediate code converter into a description instruction which can be interpreted by the description unit of an OS and outputting the converted data; and a print instruction allocator for sending a print instruction received through the description unit of the OS from the application to the intermediate data converter and sending the print instruction received through the description unit of the OS from the description instruction generator to the printing data generator, a conventional printing data converter can be employed so that the number of developing processes can be decreased when such an arrangement is provided.

Still further, since the description instruction is a GDI function, the print instruction is a DDI function and the printing data is a printer language,

00609380-103100

Still further, since the composed job generator generates the layout information of the composed job for each physical page on the basis of the layout information of a plurality of print jobs, the layout information can be provided for each physical page.

Still further, since the information processor further comprises a close arrangement instructing unit for instructing the logical pages of each print job to be closely arranged; and the composed job information generator determines to closely arrange the logical pages in the physical pages when a close arrangement is instructed by the close arrangement instructing unit, the arrangement of the logical pages excellent in its variation can be achieved. Accordingly, the print jobs can be prepared with ease as desired by the user.

Still further, since the close arrangement
instructing unit performs any one of a close
arrangement instruction for closely arranging the
logical pages on the same physical page, a back side
close arrangement instruction for compactly arranging
the logical pages on a back side when the back side of
the same physical page are unoccupied, and no
instruction for a close arrangement by constantly
changing the physical pages when original print jobs
are different, the arrangement of the logical pages
rich in its variation can be realized, the preparation
of the print jobs can be attained as desired by the
user.

Furthermore, since the information process further
includes a preview controller for controlling the
preview of the composed job to be displayed, an actual
output result can be notified to the user before a
printing operation.

00699389 103100